

REMARKS

The claims are claims 1 and 4.

Claim 1 is amended. Claims 2, 3 and 5 to 18 are cancelled. Claim 1 is amended to incorporate subject matter previously recited in cancelled claims 2 (optimizer producing optimized intermediate code) and 3 (linker produces executable code).

Claims 1, 4, 8 and 11 were rejected under 35 U.S.C. 102(e) as anticipated by Burch U.S. Patent No. 6,978,450.

Claim 1 recites subject matter not anticipated by Burch. Claim 1 recites "a compiler that receives source code and generates an object file comprising object code and intermediate code." The OFFICE ACTION cites compiler 108 illustrated in Figures 1 and 3B of Burch as producing intermediate code 122 and object code 120. The Applicants respectfully submit that this view of Burch is incorrect. Burch teaches that compiler 107 generates intermediate code 122. Burch states at column 7, lines 55 to 60 and at column 8, lines 33 to 35:

"A source compiler 107 processes a source code file 118 and thereby transforms the source code file 118 into an intermediate code file 122."

Burch clearly states that object code 120 is produced by another structure than the compiler recited in claim 1. Burch states at column 4, lines 16 and 17:

"An intermediate code generator 113 creates object code files 120"

Burch states at column 4, line 65 to column 5, line 3:

"The present embodiment includes an optimizer 109 that generates object code 120 that includes optimization changes

which may be dependent on a particular computer system 100. Further, these system-specific changes allow the optimizer 109 to generate object code 120 that is highly tailored to optimally run on a specific computer system 100."

Burch states at column 6, lines 37 to 40:

"The compilation system 108 may include the optimizer 109, the intermediate code generator 113 including the compiler tool 102, the linker 112, the loader 115, the libraries 114, and the source compiler 107."

Thus Burch clearly teaches source compiler 107 is a different structure than optimizer 109 and intermediate code generator 113. Accordingly, Burch fails to teach that the compiler generates both object code and intermediate code as recited in claim 1. Burch teaches these outputs are generated by differing structures. Accordingly, claim 1 is allowable over Burch.

Claim 1 recites further subject matter not anticipated by Burch. Claim 1 recites the linker "receives the object file comprising object code and intermediate code." The OFFICE ACTION cites linker 112, object code 120, object code stream 203 and illustrated in Figure 3A of Burch and intermediate code 122, high level scope optimizer 109!, object code 120 and executable file 124 illustrated in Figure 3B as anticipating this subject matter. Burch states at column 8, lines 8 to 17:

"The optimizer 109 may optionally operate on the intermediate code stream 122 to enhance the resulting object code file 120 for the purpose of producing an executable file 124 that executes more efficiently. The object code file 120 may preserve the execution instructions included in the intermediate code stream 122 by including execution instructions in the object code stream 203. The linker 112 subsequently generates an executable file 124 by linking the associated object code files 120 and other files such as libraries 114 (as shown in FIG. 1)."

The Applicants submit that this portion of Burch teaches that linker 112 operates upon object code files 120. Optimizer 109 operates on intermediate code stream 122 to produce an object code file 120. Thus linker 112 taught in Burch fails to receive both object code and intermediate code as required by the recitations of claim 1. Figure 3A of Burch shows intermediate code generator stream 122 produced by source code compilers 107. Linker 112 does not receive intermediate code as claimed in claim 1 but receives only object code file 120. Accordingly, claim 1 is allowable over Burch.

Claim 1 recites still further subject matter not anticipated by Burch. Claim 1 recites that the linker "provides the intermediate code to the code optimizer." Figure 3A of Burch illustrate optimizer 109 producing object code file 120 from intermediate code stream 122 without intervention of linker 112. There is no signal path between linker 112 and optimizer 109 illustrated in Figure 3A of Burch. The OFFICE ACTION fails to cite any teaching of Burch that linker 112 passes any information to optimizer 109. Figure 3B of Burch similarly illustrates high level scope optimizer 109A and low level scope optimizer 109B disposed in the process stream before linker 112. There is no signal path for linker 112 of Burch to provide intermediate code to either high level scope optimizer 109A or low level scope optimizer 109B. Accordingly, claim 1 is allowable over Burch.

Claim 4 recites subject matter not anticipated by Burch. Claim 4 recites "the linker sends only portions of the intermediate code to the code optimizer." The OFFICE ACTION cites column 3 lines 4 to 38, column 8, lines 18 to 30 and column 9, lines 3 to 23 as anticipating this subject matter. Burch states at column 4, lines 4 to 7 (within the portion cited by the Examiner):

"The compiler front-end emits intermediate code only for those

constructs that are needed, isolating the effect of an input file change to those that could potentially change the resulting object code."

This states that it is the compiler front end that selects only part of the intermediate code. This selection taught in Burch is thus not the linker recited in claim 4. This portion of Burch further fails to state that the emitted intermediate code is sent to the code optimizer as required by claim 4. As noted above Burch fails to teach any manner in which linker 112 sends intermediate code to optimizer 109. Accordingly, Burch fails to anticipate the sending of only portions of the intermediate code from the linker to the optimizer. Thus claim 4 is not anticipated by Burch.

The Applicants respectfully submit that all the present claims are allowable for the reasons set forth above. Therefore early reconsideration and advance to issue are respectfully requested.

If the Examiner has any questions or other correspondence regarding this application, Applicants request that the Examiner contact Applicants' attorney at the below listed telephone number and address to facilitate prosecution.

Texas Instruments Incorporated
P.O. Box 655474 M/S 3999
Dallas, Texas 75265
(972) 917-5290
Fax: (972) 917-4418

Respectfully submitted,

/Robert D. Marshall, Jr./
Robert D. Marshall, Jr.
Reg. No. 28,527